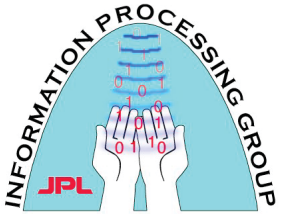
National Aeronautics &
Space Administration

Frame Synchronization without Attached Sync Markers

Jon Hamkins

Jet Propulsion JPL
California Institute of Technology

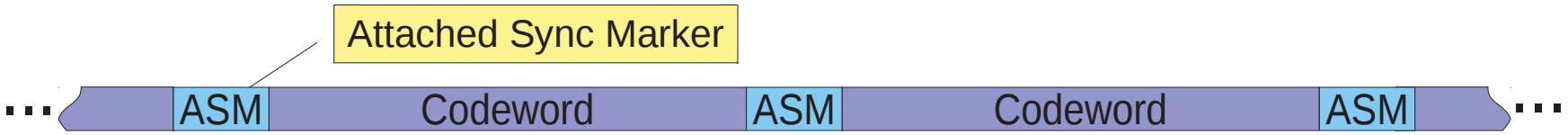
March 9, 2011



Introduction

Conventional frame synchronization

- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



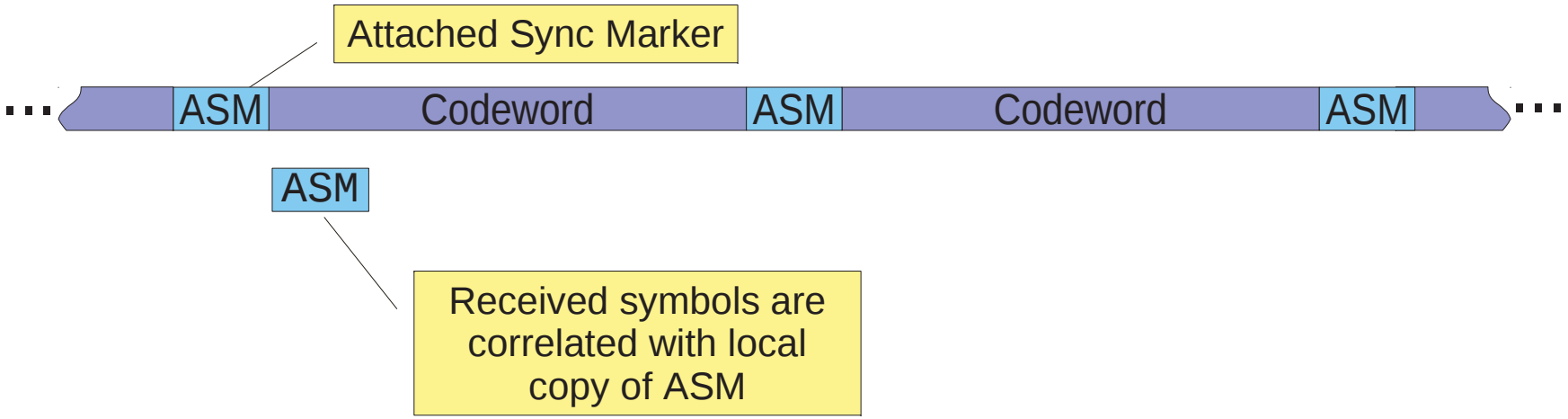
- For CCSDS low-density parity-check (LDPC) codes, the ASM is the 64-bit pattern:

0000001101000111011101101100011100100111001010001001010110110000

Introduction

Conventional frame synchronization

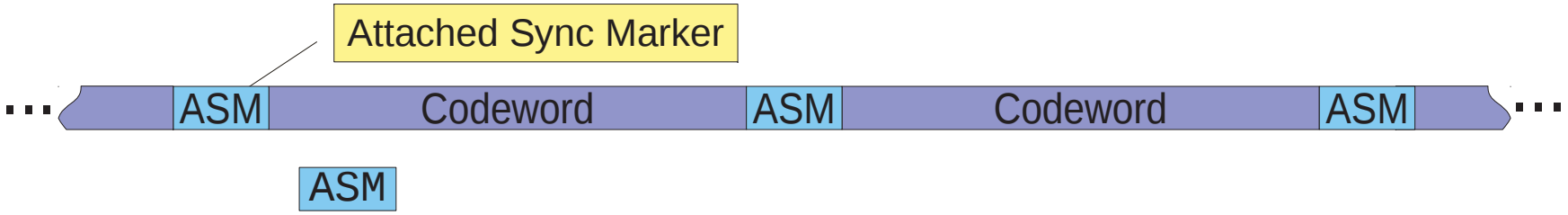
- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



Introduction

Conventional frame synchronization

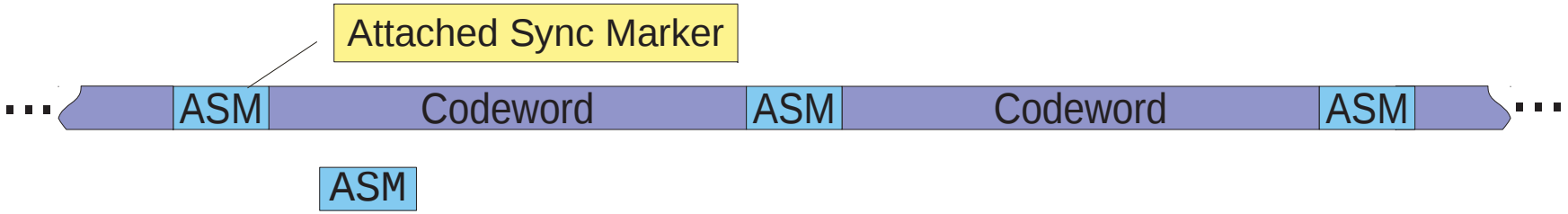
- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



Introduction

Conventional frame synchronization

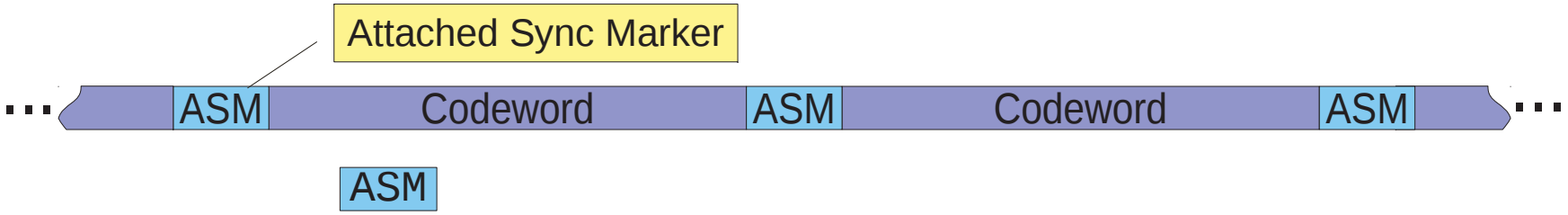
- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



Introduction

Conventional frame synchronization

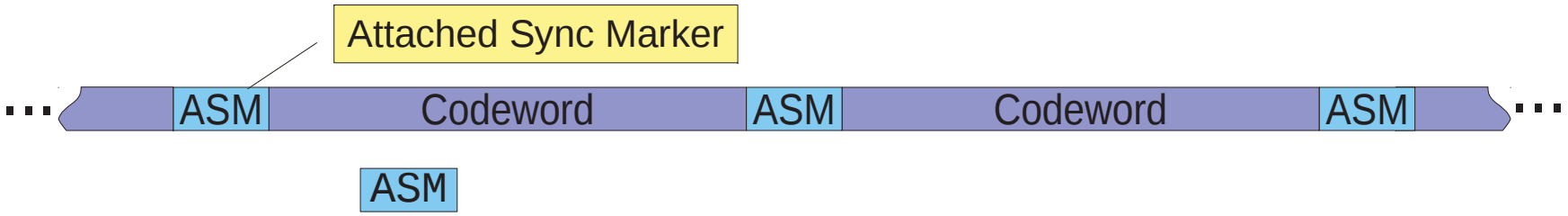
- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



Introduction

Conventional frame synchronization

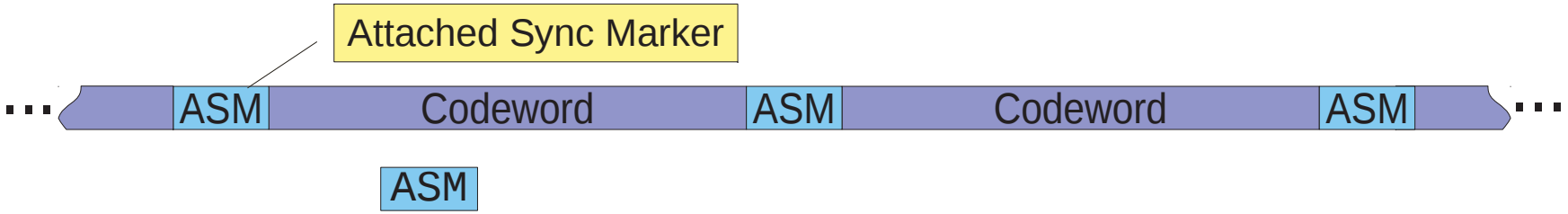
- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



Introduction

Conventional frame synchronization

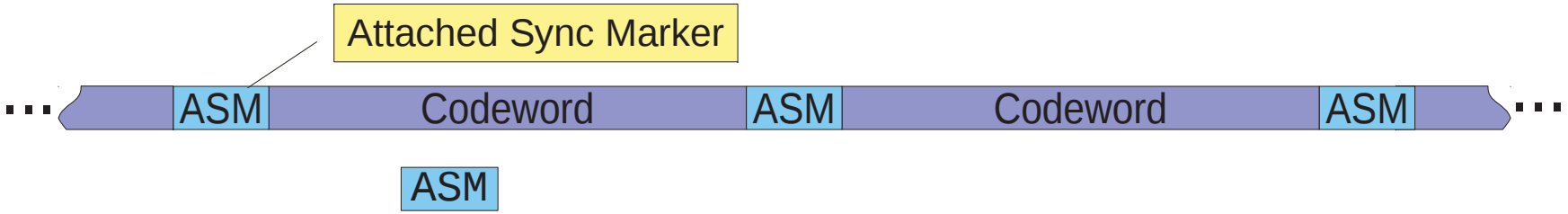
- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



Introduction

Conventional frame synchronization

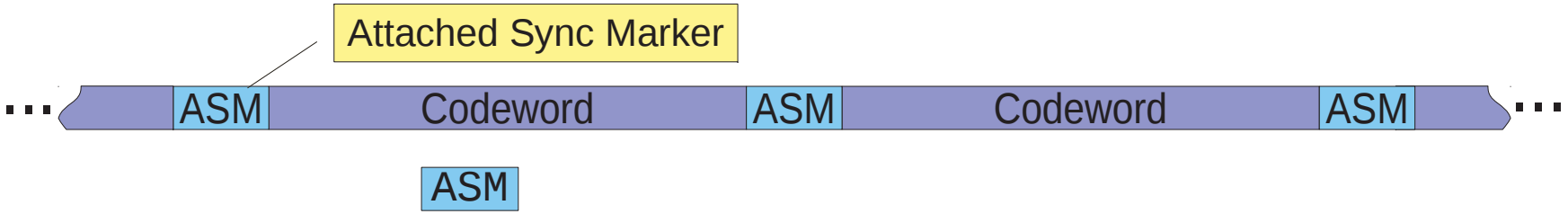
- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



Introduction

Conventional frame synchronization

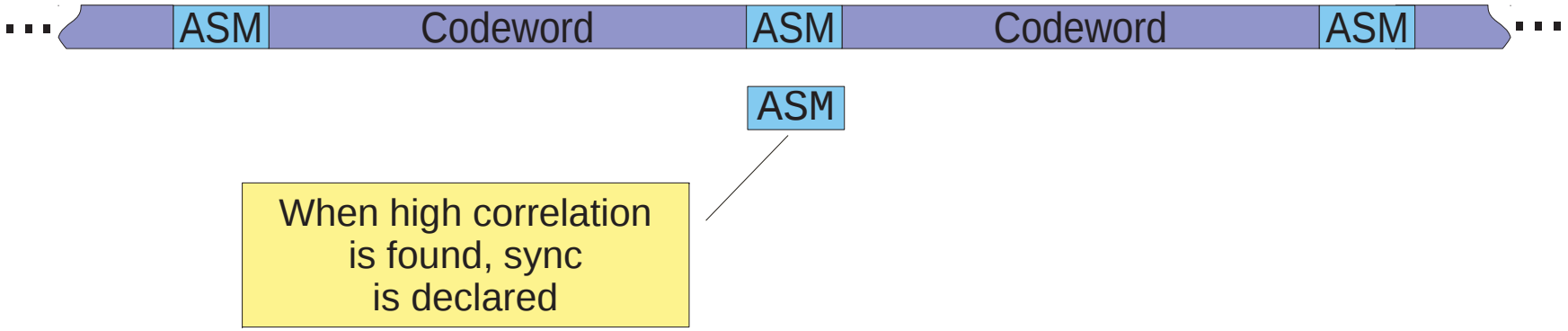
- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



Introduction

Conventional frame synchronization

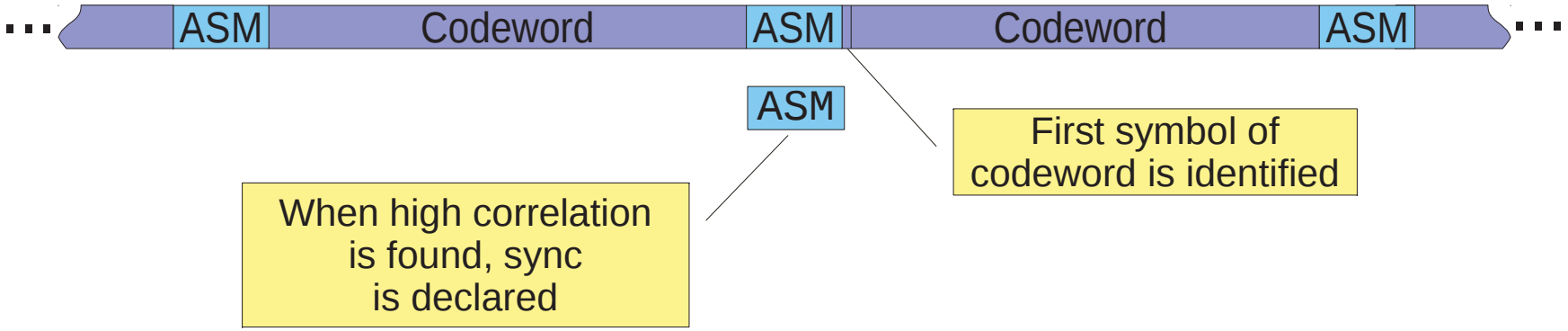
- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



Introduction

Conventional frame synchronization

- *Attached Sync Markers* (ASMs) are inserted between codewords
- ASM+codewords are sent one after the other, without gaps:



- This method has been used successfully for decades for legacy codes
- It was also successfully tested for the emerging LDPC code standards

National Aeronautics & Space Administration

Overhead of ASMs

Overhead of including ASM, for CCSDS codes:

	Code	Transmitted ASM Length	Codeword Length	E_b/N_0 penalty (dB)
Standard downlink codes	Reed-Solomon	32	2040	0.1
	RS+CC	64	>4080	<0.1
	Turbo	192	>3568	<0.1
	LDPC	64	>2048	<0.1
	LDPC	64	≤2048	0.1 to 0.2

National Aeronautics & Space Administration

Overhead of ASMs

Overhead of including ASM, for CCSDS codes:

	Code	Transmitted ASM Length	Codeword Length	E_b/N_0 penalty (dB)
Standard downlink codes	Reed-Solomon	32	2040	0.1
	RS+CC	64	>4080	<0.1
	Turbo	192	>3568	<0.1
	LDPC	64	>2048	<0.1
	LDPC	64	≤2048	0.1 to 0.2
Standard uplink code	BCH	80*	64**	3.5

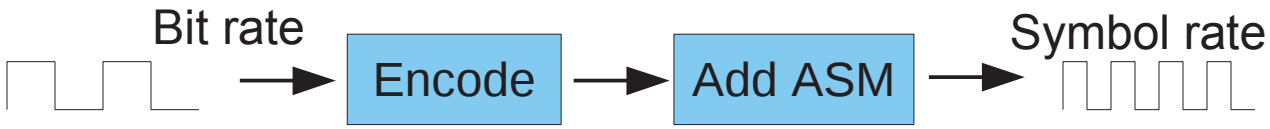
Eliminating the ASM would significantly increase uplink coding gain.

* 16-bit marker and 64-bit tail sequence
** Assuming 1 ASM per codeword (the minimum acqusition-time configuration)

Clock Distribution

National Aeronautics & Space Administration

In a hardware implementation, clocks run at the ***bit rate*** and ***symbol rate***:



To avoid buffering, it is helpful if the symbol rate / bit rate is a simple ratio.

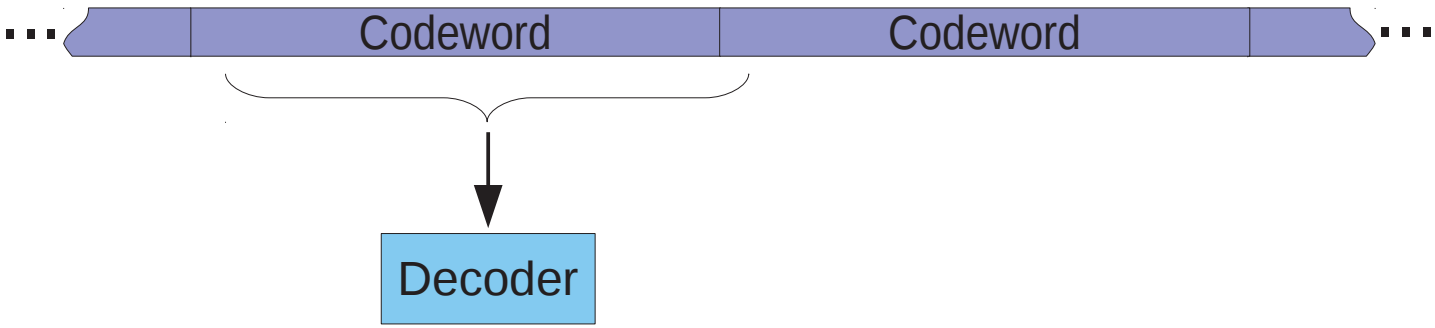
Here are the ratios for CCSDS standard LDPC codes:

Input length (bits)	Rate	ASM length	Symbol rate to bit rate ratio (including ASM)	Symbol rate to bit rate ratio (without ASM)
1024	1/2	64	33 : 16	2 : 1
4096	1/2	64	129 : 64	2 : 1
16384	1/2	64	513 : 256	2 : 1
1024	2/3	64	25 : 16	3 : 2
4096	2/3	64	97 : 64	3 : 2
16384	2/3	64	385 : 256	3 : 2
1024	4/5	64	21 : 16	5 : 4
4096	4/5	64	81 : 64	5 : 4
16384	4/5	64	321 : 256	5 : 4

Eliminating the ASM would simplify clock distribution.

National Aeronautics & Space Administration

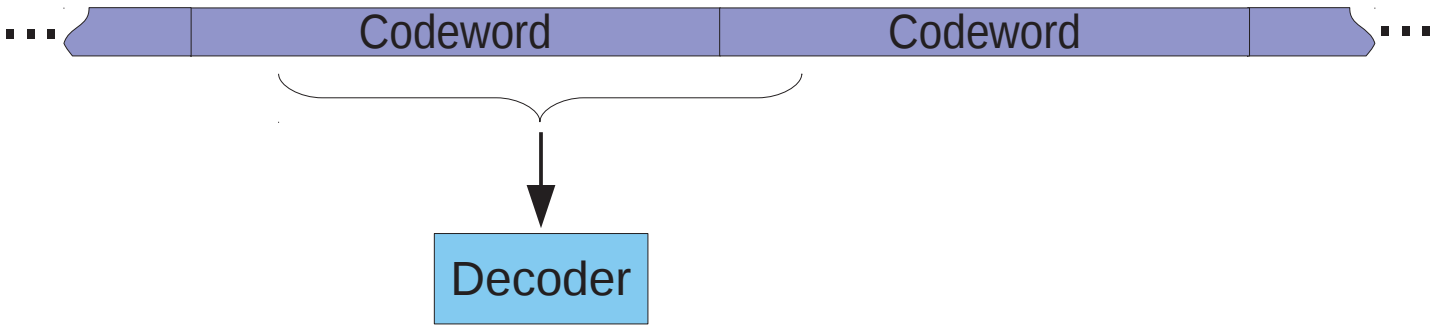
New Frame Sync Approach



- Eliminate ASMs from transmission – just transmit codewords
- Attempt to decode at every possible offset
- When correct decoding results, frame sync has been found

National Aeronautics & Space Administration

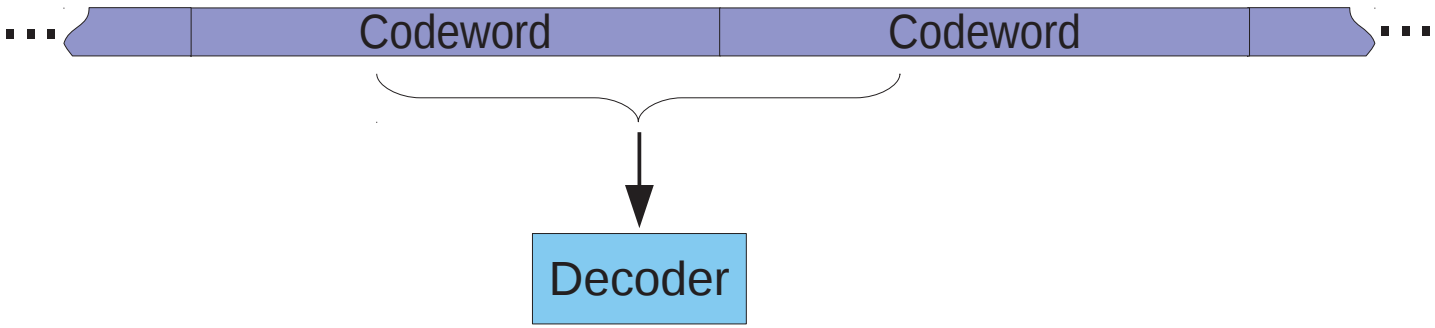
New Frame Sync Approach



- Eliminate ASMs from transmission – just transmit codewords
- Attempt to decode at every possible offset
- When correct decoding results, frame sync has been found

National Aeronautics & Space Administration

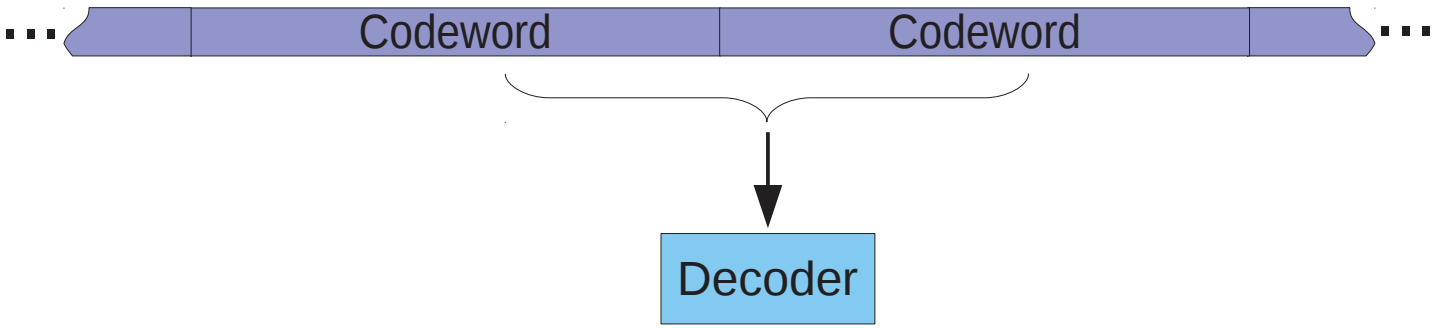
New Frame Sync Approach



- Eliminate ASMs from transmission – just transmit codewords
- Attempt to decode at every possible offset
- When correct decoding results, frame sync has been found

National Aeronautics & Space Administration

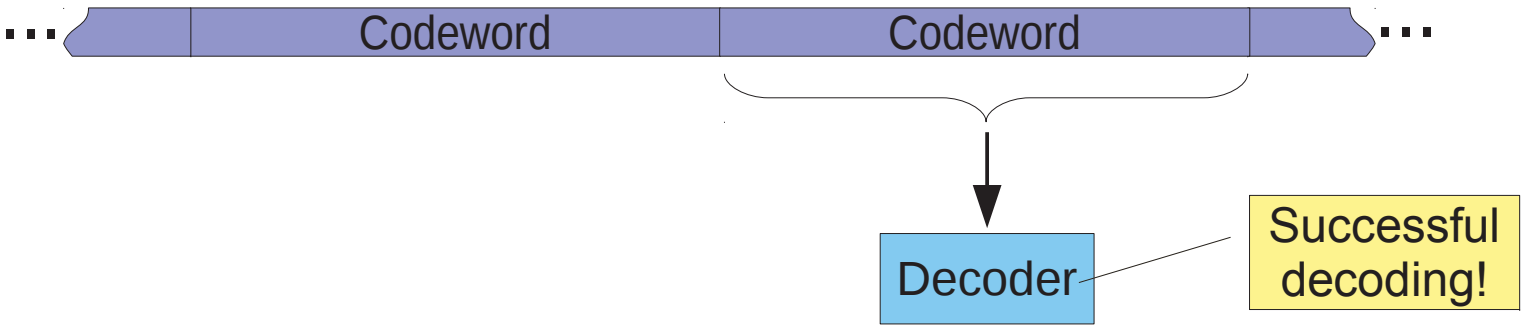
New Frame Sync Approach



- Eliminate ASMs from transmission – just transmit codewords
- Attempt to decode at every possible offset
- When correct decoding results, frame sync has been found

National Aeronautics & Space Administration

New Frame Sync Approach

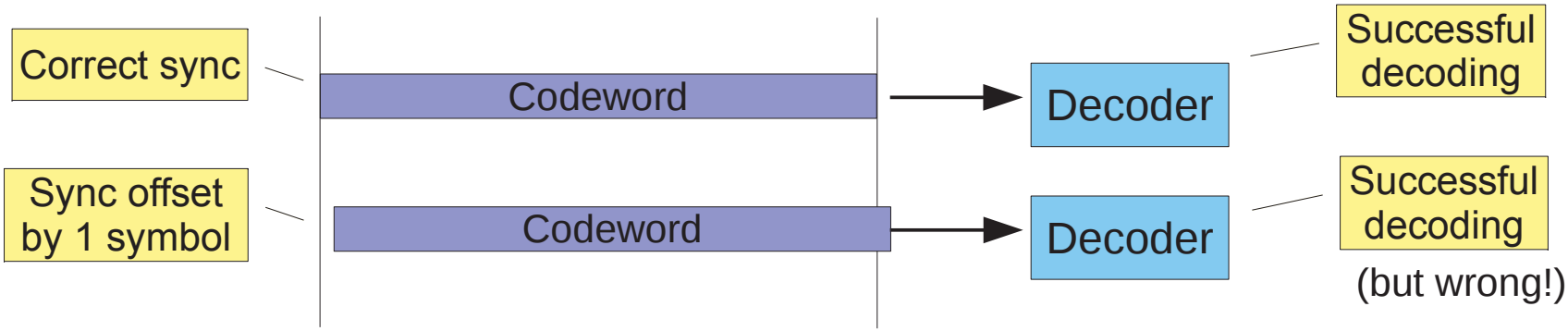


- Eliminate ASMs from transmission – just transmit codewords
- Attempt to decode at every possible offset
- When correct decoding results, frame sync has been found

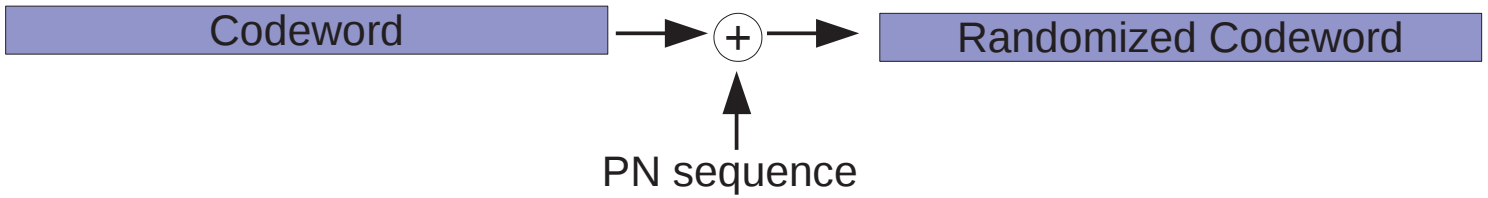
Solution to a False Sync Problem

National Aeronautics & Space Administration

- Problem: The CCSDS LDPC codes are quasi-cyclic. A cyclic shift by 1 symbol is still decodable (and to a wrong codeword!):



- Solution: Use the CCSDS-recommended randomizer at transmitter:



- Randomized codewords
  - Do not have the quasi-cyclic property
  - Do not falsely decode at incorrect offset
- Conclusion: use the randomizer when using the new sync method

National Aeronautics &
Space Administration

New Frame Sync Approach

Properties of the new approach:

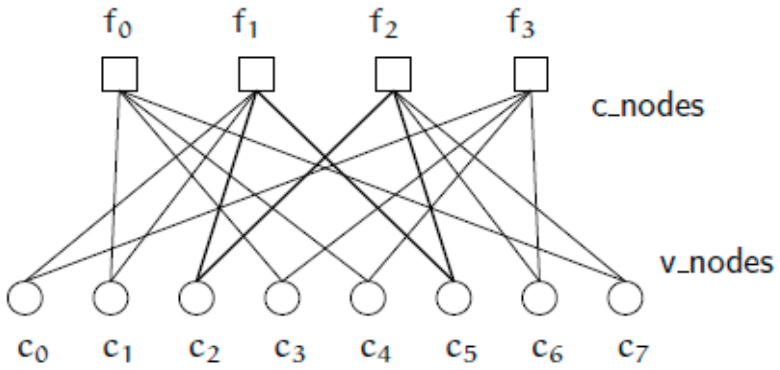
- All ASM transmissions are eliminated
- The otherwise-idle decoder is now utilized during synchronization
- Algorithm is guaranteed to find correct offset whenever decodable data is present
- Up to *n* offsets must be tested, where *n* is the codeword length

A Faster Version of the Synchronizer

Attempting full decoding at every offset can take a while. Can we make it faster?

With LDPC and turbo codes:

- Decoding consists of a series of iterations
- After each iteration, each code symbol is assigned a log-likelihood ratio (LLR), relating the probability that the symbol is a 0 or a 1
- A properly synchronized codeword will converge fundamentally differently from an improperly synchronized codeword



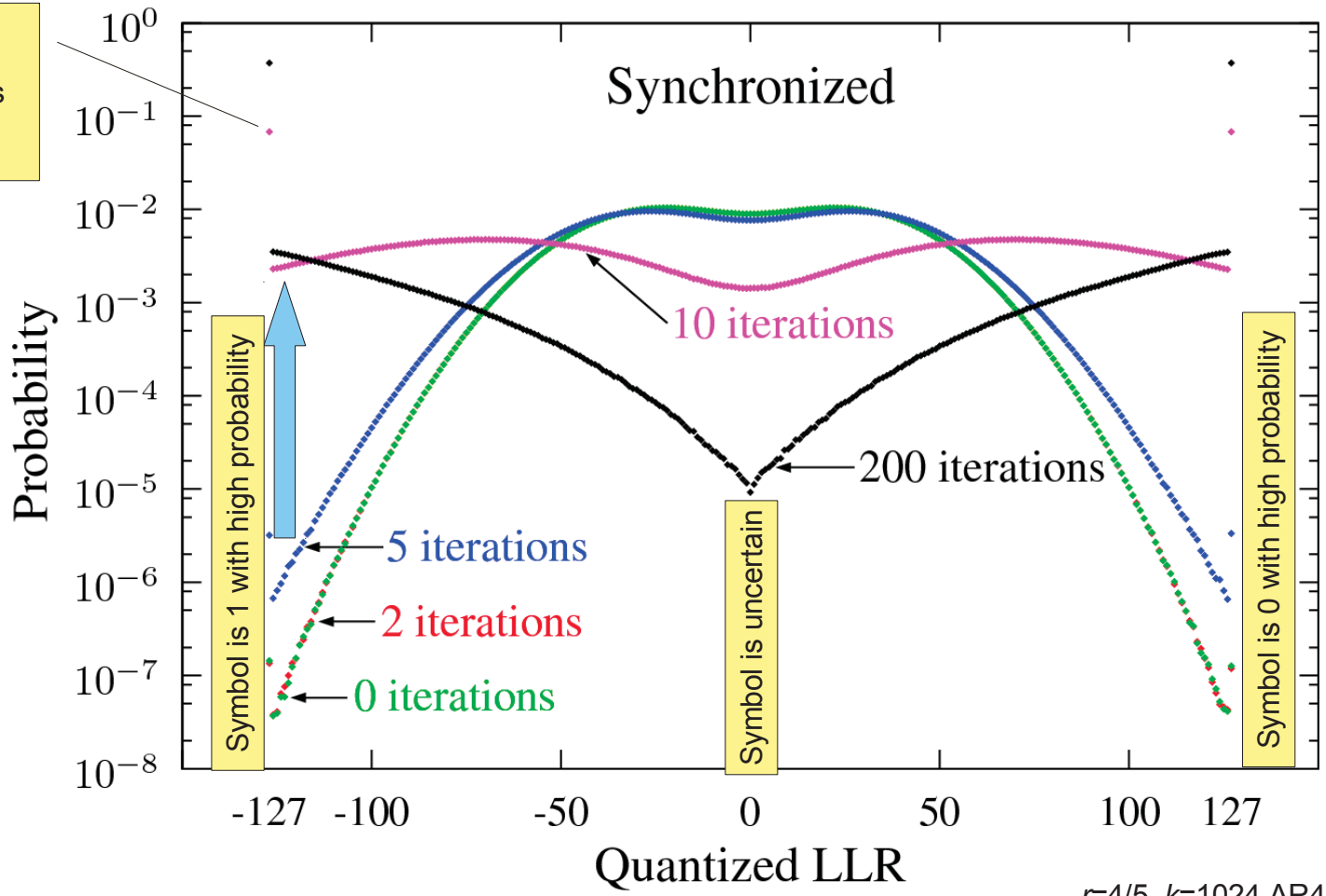
Idea:
Use only a ***few*** decoder iterations and develop a metric to distinguish the correct and incorrect sync states.

Developing a Metric, Using Variable Nodes

National Aeronautics & Space Administration

Distribution of *variable node LLRs*, when correctly synchronized:

After 10 iterations ***many*** nodes have high abs LLRs

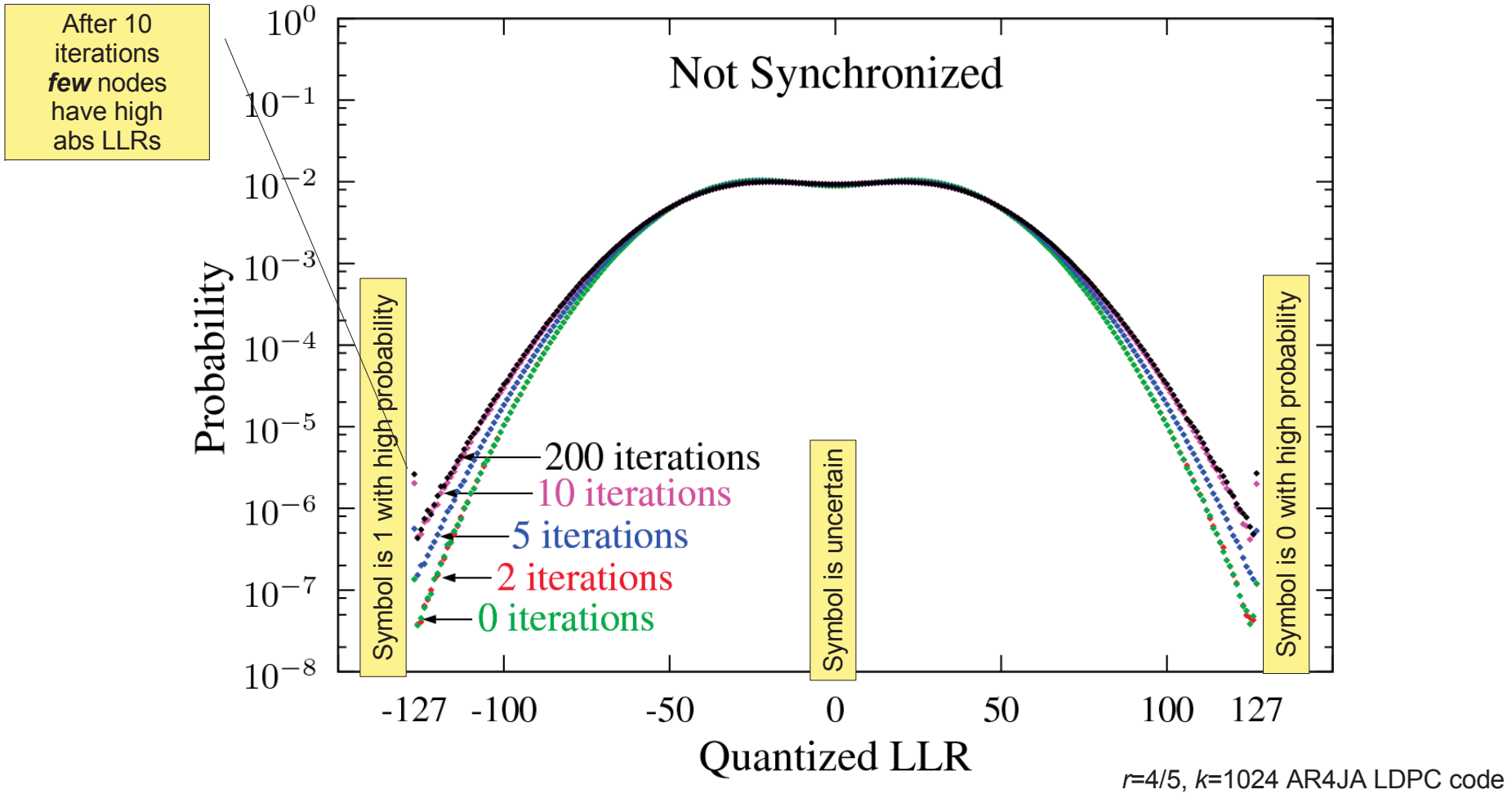


r=4/5, *k*=1024 AR4JA LDPC code

Developing a Metric, Using Variable Nodes

National Aeronautics & Space Administration

Distribution of *variable node LLRs*, when incorrectly synchronized:



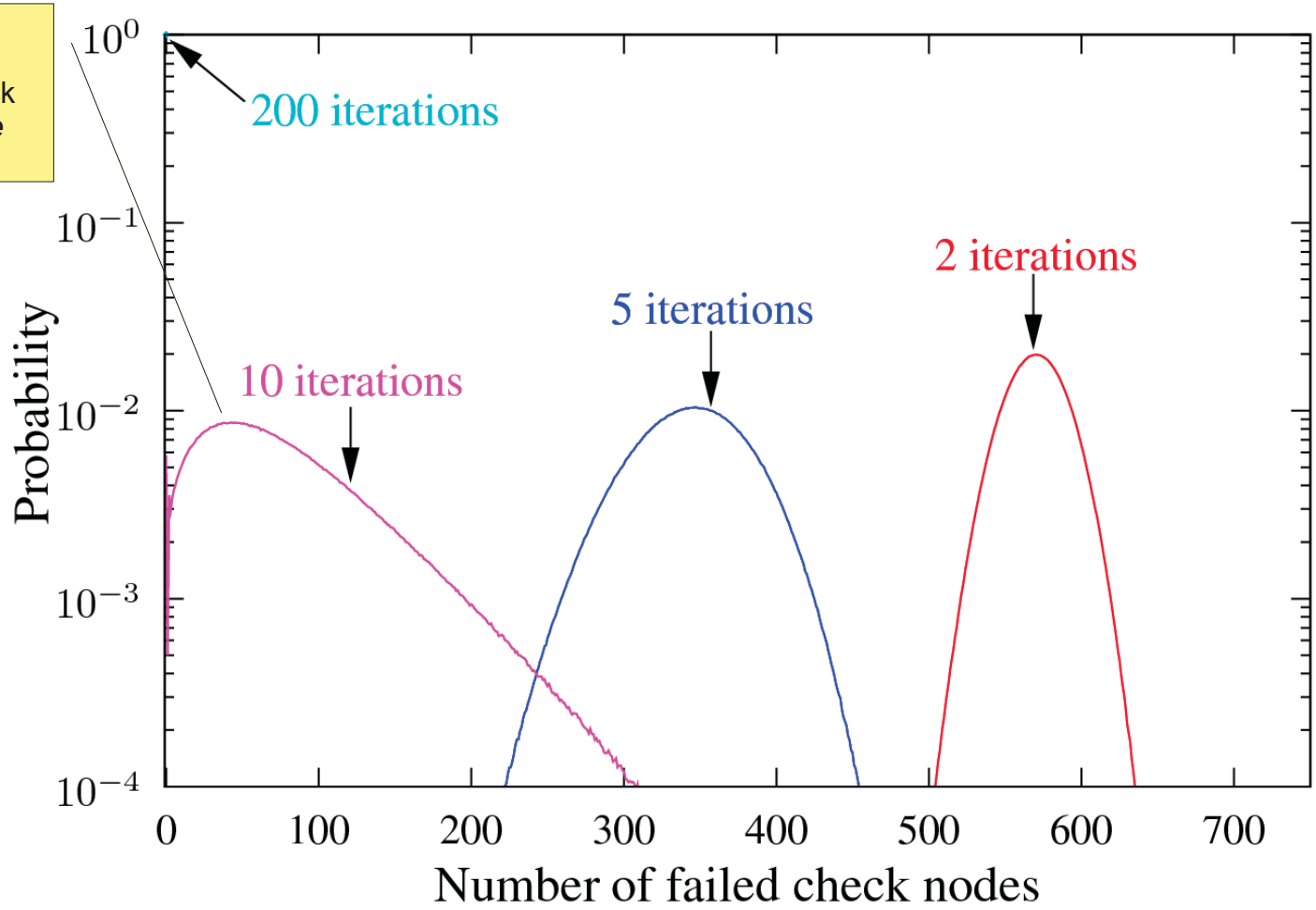
r=4/5, *k*=1024 AR4JA LDPC code

Developing a Metric, Using Check Nodes

National Aeronautics & Space Administration

Distribution of *number of satisfied check nodes*, when correctly synchronized:

After 10 iterations most check nodes ***are*** satisfied



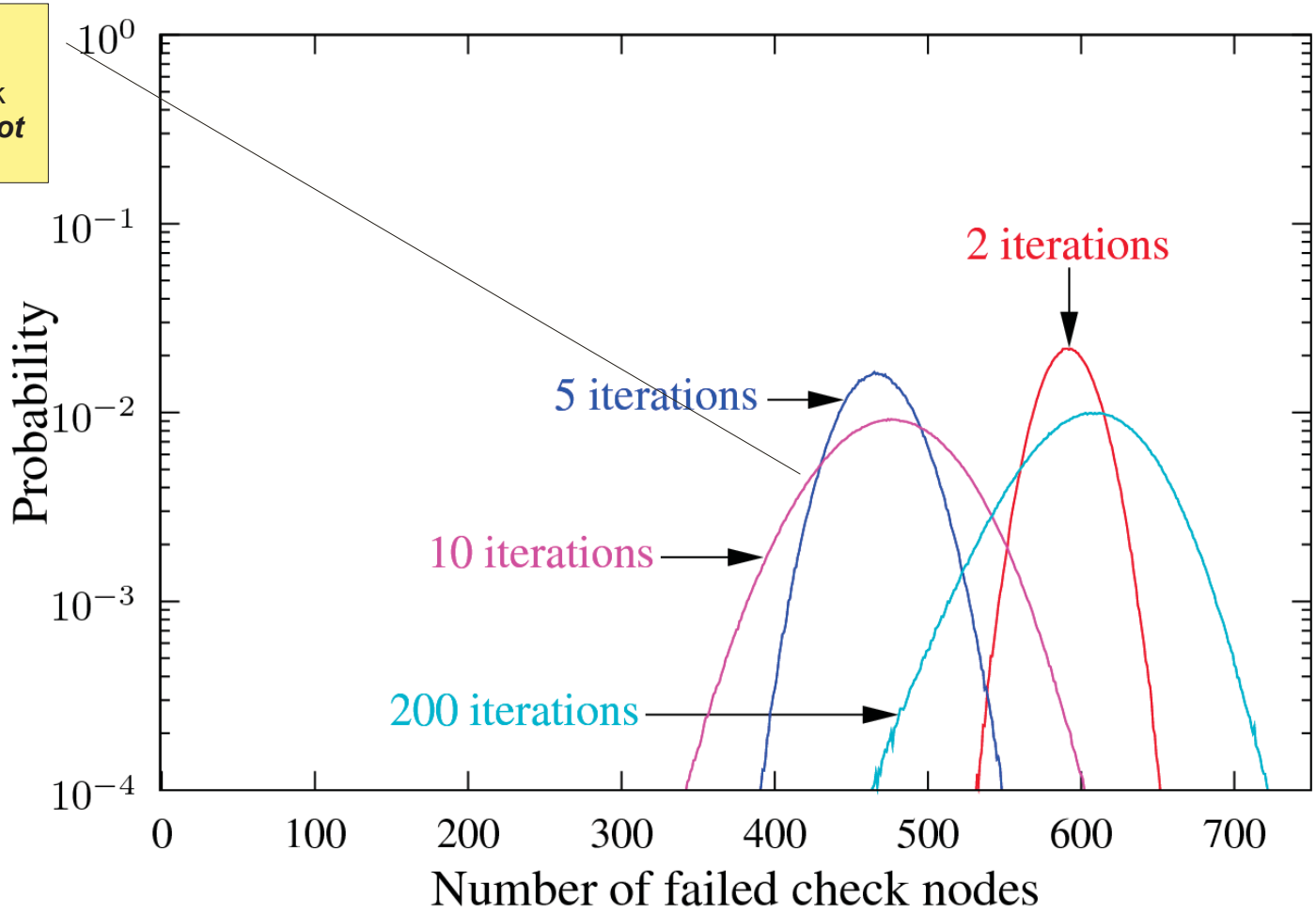
r=4/5, *k*=1024 AR4JA LDPC code

Developing a Metric, Using Check Nodes

National Aeronautics & Space Administration

Distribution of *number of satisfied check nodes*, when not synchronized:

After 10 iterations most check nodes are ***not*** satisfied



r=4/5, *k*=1024 AR4JA LDPC code

National Aeronautics & Space Administration

Possible Metrics

- Metric for variable nodes:

$$M = \sum_{i=1}^{n} f(\lambda_i)$$

where λ_i is the ith LLR and *f*(.) is an even, monotonically increasing function

  - Reasonable choices:

1. $f(x) = |x|^a$, for some real positive a
2. $f(x) = e^{|x|}$
3. $f(x) = \log(1 + |x|)$
4. $f(x) = I_{\{|x| \geq \eta\}}$, where I is the indicator function and η is a threshold

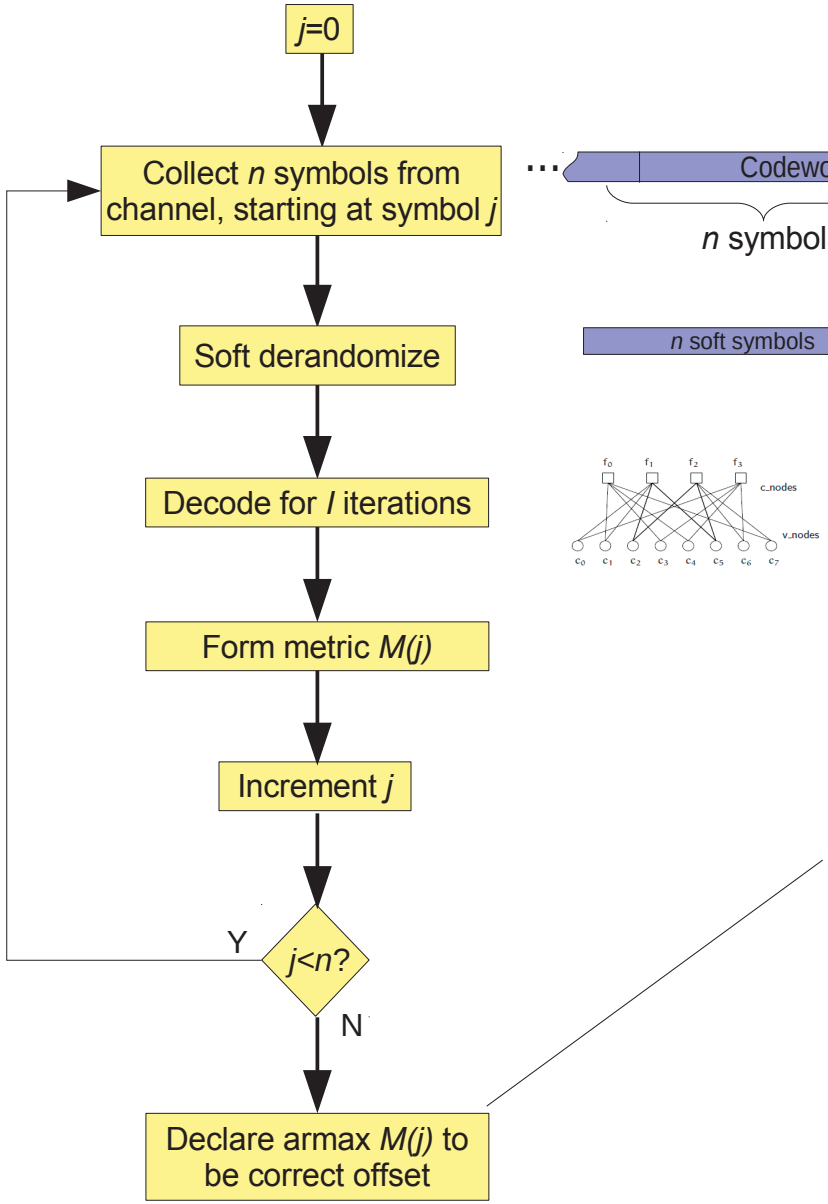
- Metric for check nodes:

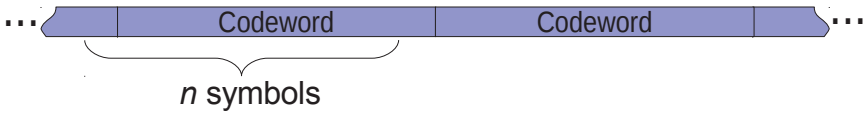
$$M = \sum_{i=1}^{n-k} I_{\{\text{check node } i \text{ satisfied}\}}$$

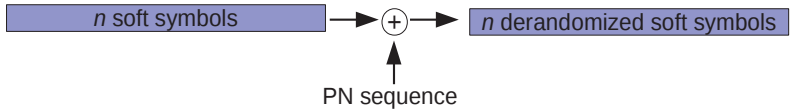
i.e., count the number of satisfied check nodes

New Frame Sync Algorithm

National Aeronautics & Space Administration



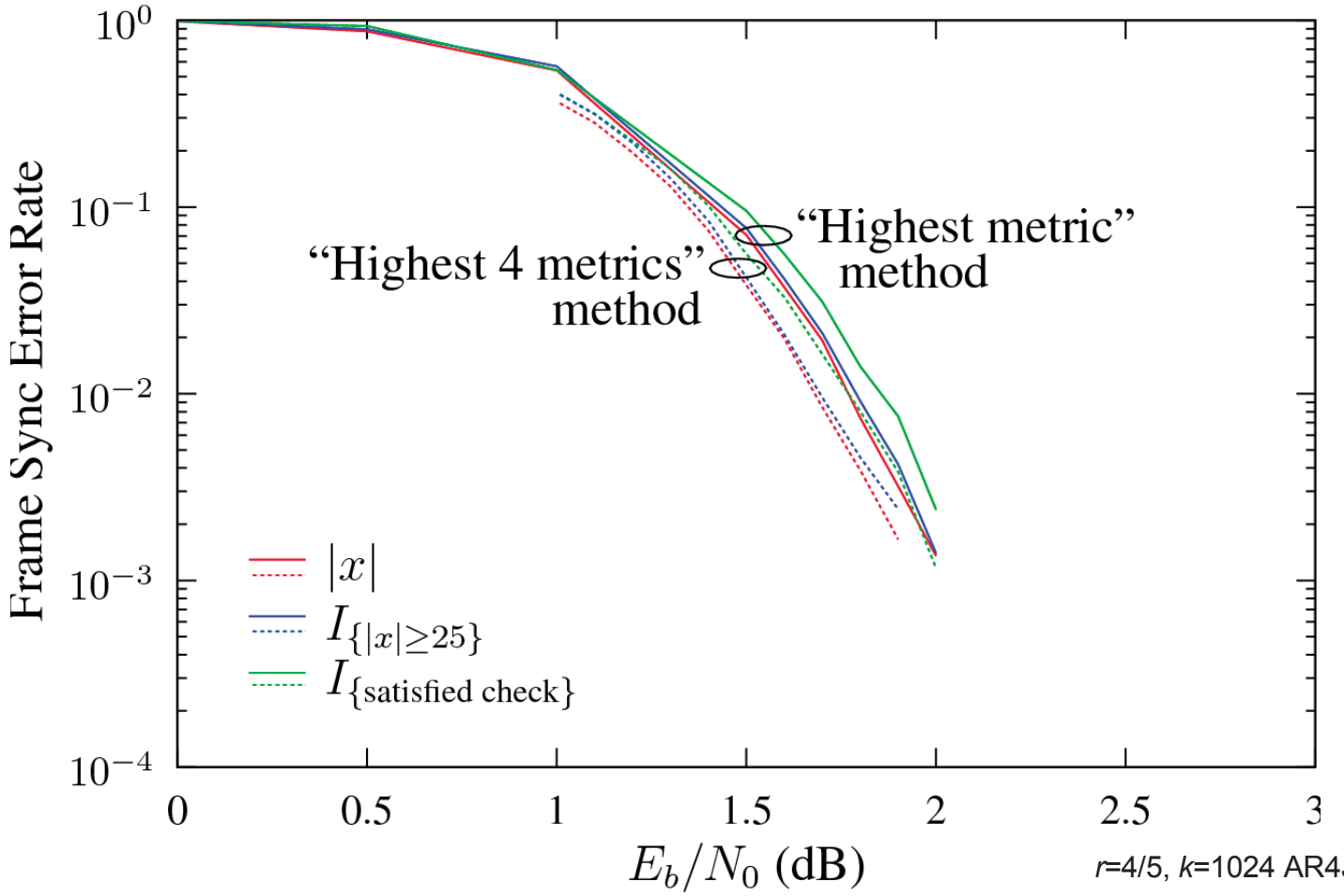




At this point, full decoding can occur.
If this fails, the sync process may be repeated.

Or, full decoding may be attempted at the offset of the 2nd, 3rd, etc. highest metric

Frame Sync Performance

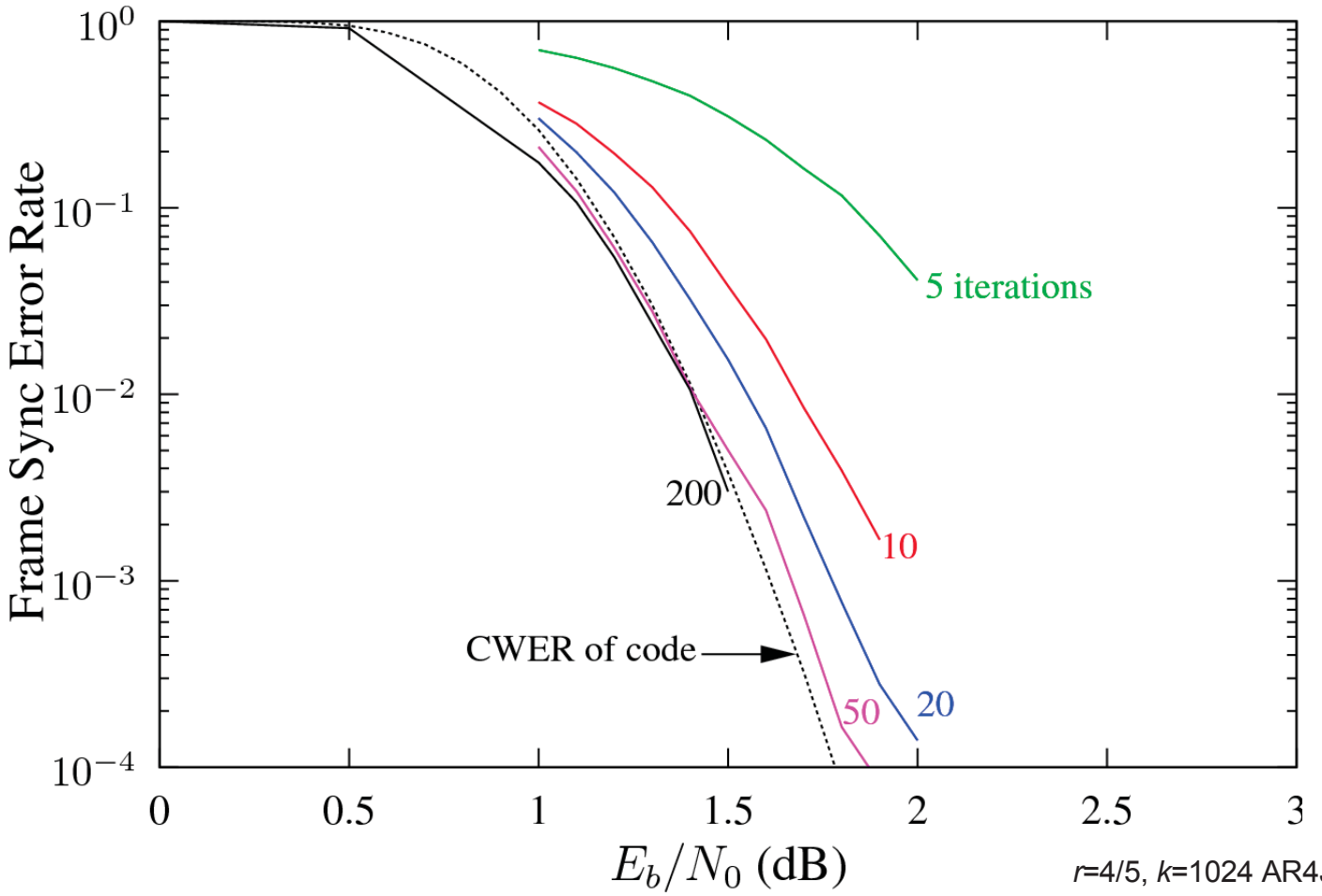


r=4/5, *k*=1024 AR4JA LDPC code

Performance is insensitive to choice of metric

Conclusion: pick a simple-to-compute metric (|*x*|)

Sync Performance: Based on 1 Codeword

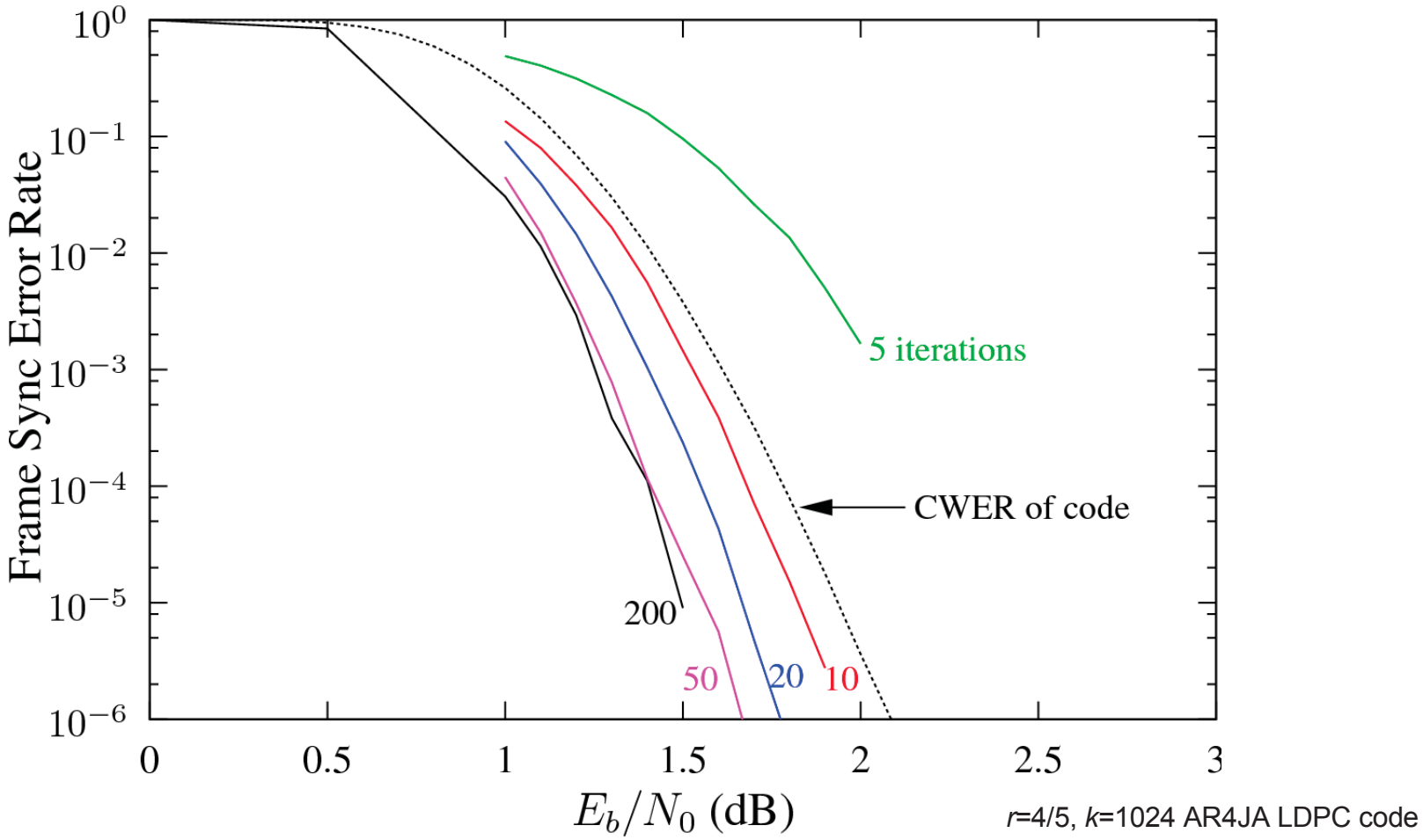


r=4/5, *k*=1024 AR4JA LDPC code

Sync performance is not as good as code performance.

Conclusion: Adequate sync is not achieved within one codeframe length.

Sync Performance: Based on 2 Codewords



r=4/5, *k*=1024 AR4JA LDPC code

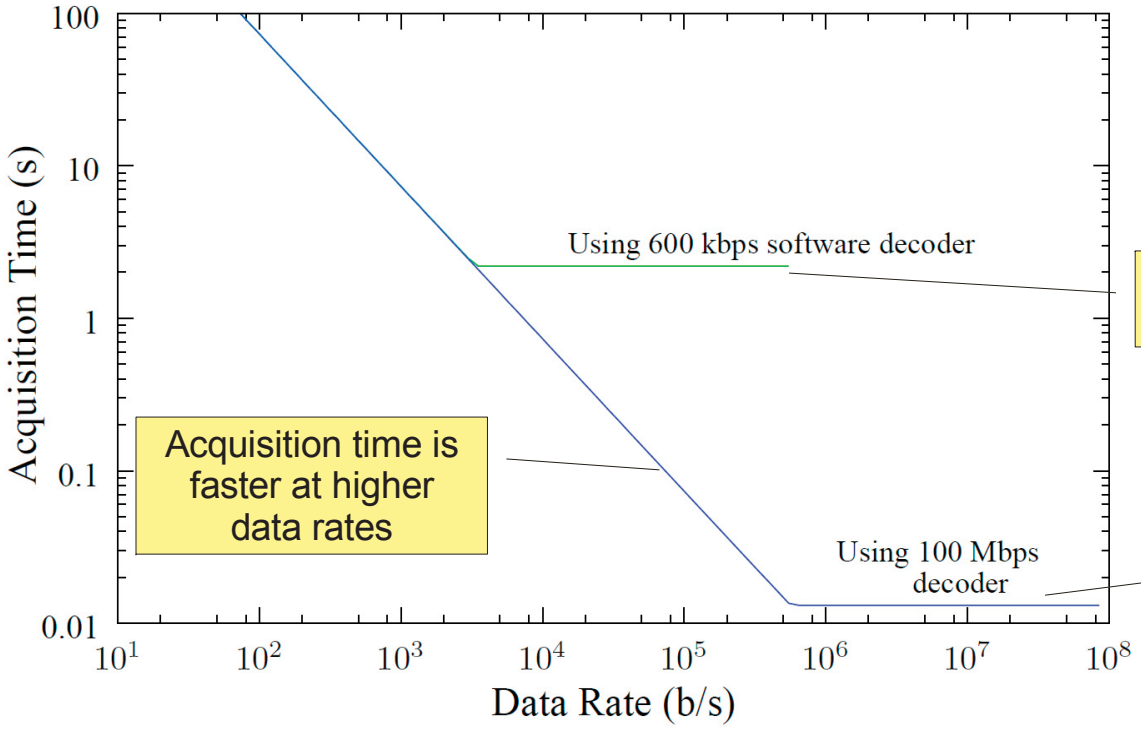
For >10 iterations, sync performance is better than code performance.

Conclusion: Good sync error rate is achieved within two codeframe lengths.

National Aeronautics & Space Administration

Acquisition Time

Decoder acquisition time is reasonably good:



Acquisition time is faster at higher data rates

Acquires in ~2 s, for data rates > 10 kbps

Acquires in ~15 ms, for data rates > 500 kbps

r=4/5, *k*=1024 AR4JA LDPC code

National Aeronautics & Space Administration

Conclusions

- A new frame synchronizer was presented
  - Eliminates need to transmit attached sync markers (ASMs)
  - In ~10 iterations, decoder can distinguish between sync and non-sync states
- Advantages:
  - 3.5 dB coding gain for standard uplink codes
  - 3 dB coding gain for proposed uplink LDPC codes of similar length
  - 0.2 dB coding gain for CCSDS LDPC codes
  - Simplified clocking in hardware on spacecraft and on ground